(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) Publication number: **0 524 362 A1**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: **91402072.2**

(61) Int. Cl.5: **G06F 3/14**

(22) Date of filing: **24.07.91**

(43) Date of publication of application:
27.01.93 Bulletin 93/04

(84) Designated Contracting States:
DE ES FR GB IT NL

(71) Applicant: **TEXAS INSTRUMENTS FRANCE**
B.P. 5
F-06270 Villeneuve Loubet(FR)
(84) **FR**

(71) Applicant: **TEXAS INSTRUMENTS INCORPORATED**
13500 North Central Expressway
Dallas Texas 75265(US)

(84) **DE ES GB IT NL**

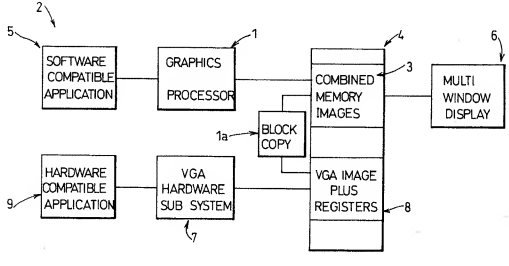
(72) Inventor: **Tannyeres, Louis**
Les Pugets - Rue Jean Gionnot - BAT.D
06700 St.Laurent Du Var(FR)
Inventor: **Goddard, Marc**
Le Ceres, Avenue Elise
F-06140 Vence(FR)

(74) Representative: **Habasque, Etienne Joel Jean-François et al**
Cabinet Lavoix 2, Place d'Estienne d'Orves
F-75441 Paris Cédex 09(FR)

(54) Display adapter.

(57) This adapter is connected between a host processor (2) and a display unit (6) of a computerized tool.

It comprises a graphics processor (1) connected between the host processor and a first part (3) of a video memory (4) associated with the display unit and a logic based hardware sub-system (7) connected between said host processor (2) and a second part (8) of said video memory. Means (1a) for deriving displays from the first part and/or the second part of the memory are also provided.



FIG·1

EP 0 524 362 A1

The present invention relates to a display adapter. Display adapters are used intermediate data to be displayed and the display itself.

Adapters may be used in all areas of video displays used in computing applications such as personal computers, workstations, graphic terminals, printers, etc ...

More particularly, an adapter is connected for example between a host processor and a display unit of a personal computer or of another computerized tool in order to manage the display unit in accordance with control signals from the host processor.

Currently, display systems use either non-processor based display adapters such as VGA (Video Graphics Arrays) which contain only low-level logic functions and registers and require that the host processor application software or operating system environment perform essentially all of the display generation and manipulation of graphics processor based adaptors such as TIGA (Texas Instruments General Architecture) based boards which interface via a high-level language or command list system. Further details of arrangements of the latter type are to be found in "Texas Instruments Graphics Architecture User's Guide", 1989; "TMS34010 User's Guide", August 1988; and "TIGA-340 Interface" all of which one documents currently available to the public from Texas Instruments Incorporated. Reference may also be made to United States Patent 4,752,893.

In the past, and indeed until fairly recently, only the 'dumb' register based display adapters were available. Although some firmware was available to interface these adapters via software calls (the BIOS extensions), it was too slow and cumbersome to be used for advanced high performance applications. This limitation resulted in most application programs accessing directly to the registers and frame buffer of the display hardware.

Upon the advent of higher performance hardware, either at the host processor (CPU) level or at the display adapter level it became possible to rethink the standard display interface and higher level display environments such as Microsoft Windows® started to appear. This was further encouraged by the need for effective multitasking where several application programs need to interface to the user at the same time but with total independence from each other.

Unfortunately, to fully take advantage of these display environments application programs had to be written especially for them - or at least an interface 'driver' be generated which performed the link. In addition, the old application was used to having the entire machine, including display adapter all to itself; something which is, of course, not possible in a multitasking environment.

Existing application software was therefore a priori incompatible with the new graphics environment trends.

Some compromise was offered by emulating the hardware model of the old logic adapter through a combination of software and the existing hardware. This, although providing a small consolation was not entirely satisfactory since the poor performance of such a system, again using Microsoft Windows as an example, limits its use to text mode displays only. The more useful graphics modes are not displayable in a window and the user must revert to full screen single task operation. Any software using EGA (Enhanced Graphics Arrays) or VGA graphics is therefore incompatible with multitasking multi-window systems which require that all display accesses are handled by the host Processor software. If an 'old application' is executed which requires unique control of the display system, then currently the displays from the multi-window manager is suspended and the 'old-application' takes control of the full screen display thereby removing the advantages of a multi-window user environment.

An aim of the invention is to overcome these drawbacks.

Accordingly the invention in one aspect thereof relates to a display adapter connected between a host processor and a display unit of a computerized tool, wherein a graphics processor is connected between the host processor and a first part of a memory associated with the display unit and a logic based hardware sub-system is connected between said host processor and a second part of said memory and comprising means for deriving displays from the first part and/or the second part of the memory.

A display system including a display adapter in accordance with the present invention may use both a low-level hardware register, based logic sub-system and also a graphics processor. This allows the generation of a multi-window display wherein both high level (GSP for example) and low level (VGA for example) applications can be executed.

The invention also permits such applications to be run simultaneously and further permits a merged display of data from such applications.

The system operates for example by allowing 'new' applications and display environments to interface, via their special driver routines for example, to the on-board graphics processor. The latter receives high level commands and performs the graphics execution. Because the logic-based hardware subsystem is totally independant of the graphics processor, it is available for use by the 'old' register/logic based

applications. By allocating memory in separate parts to firstly the high level graphics processor based tasks and secondly to the low level hardware based tasks both can execute concurrently in the same system and can even have totally different memory uses, display formats, register values, and so on. Forming the eventual display may then be performed by software. This may be executed by a subroutine of the graphics processor, and may, in one example, be a transfer for example a block copy from the low level memory part into some locations of the high level memory part of video memory associated with the display.

The display may then be derived form the first memory part only. Alternatively the second or both parts may be used.

One advantage of this system appears immediately in that the data transfered can be significantly processed during the transfer allowing many different storage formats and techniques for the 'old' applications, converted to 'real' displayable format by graphics processor software. In the TIGA/VGA example, this is particularly useful to allow conversion from VGA planar organisation to the packed pixel organisation of TIGA.

The hardware logic sub-system associated therewith a number of internal registers programmed by the application program. All of these register values may also stored in the common memory along side the display information greatly to facilitate task switching between 'old' applications by simply switching the local area of memory allocated to each task and allow several image areas in memory to exist at the same time allowing multiple hardware VGA windows on the same display. Base address registers in the VGA sub-system control the area of memory to be used by each task. The multi-tasking operating system would be responsible for maintaining these registers in cooperation with the software running on the associated graphics processor.

The advantage of such an arrangement is that existing software can be executed by the user in a multi-window environment without loosing the multi-window display . In addition, several hardware compatible applications can be viewed at the same time without falling back to a full-screen display for each task. Software in the host processor or associated graphics processor can decide how much, if any, of the hardware generated display is copied into the relevant window and where such a window appears on the final display. The data itself can also be manipulated in a variety of ways during the copy from the hardware image zone to the multi-window display zone. Such manipulation would take care of differing plane depths, text size, palettes, etc ... which would otherwise cause incompatibility between the two otherwise independant displays.

The hardware based sub-system is not limited to VGA compatibility but can be extended to cover other hardware based display standards such as 8514A, Hercules, and basically any situation which requires the concurrent execution of 'old' application software which expects to have entire ownership of the display system and 'new' applications designed for multitasking environments which do not have as 'exclusive' tastes.

The invention will be better understood in view of the drawings :

- Fig. 1 is a block diagram of an embodiment of a display adapter according to the invention.
- Fig. 2 is a block diagram of a card comprising an adapter according to the invention.
- Fig. 3 illustrated the GSP and VGA addressing.
- Fig. 4 illustrates the functions integrated in a logic based hardware subsystem.
- Fig. 5 illustrated the logic based hardware subsystem internal architecture.

As shown in Fig. 1, a display adapter according to the invention comprises a graphics processor 1 such as a TMS34010 connected between a host processor 2 of a computerized tool and a first part 3 of a video memory 4.

More specifically, the graphics processor is associated with at least one software compatible application 5 running on the host processor, and the first part 3 of the video memory 4 is associated with a display unit 6 as for example a muti window display unit.

The display adapter according to the invention also comprises a logic based hardware sub-system 7 as for example a VGA hardware sub-system, connected between the host processor 2 and a second part 8 of the video memory 4.

More specifically the hardware sub-system 7 is associated with at least one hardware compatible application 9 running on the host processor 2 and the second part 8 of the video memory 4 comprises VGA image and registers.

Merging means 1a are provided between the second part 8 and the first part 3 of the memory in order to transfer in the first part, the image data stored in the second part so that combined memory images are stored in this first part of the memory and depicted on the display unit order the control of the graphics processor 1.

In one case, these merging means comprises means for copying data from the low level memory part 8 into some location of the high level memory part 3.

As explained before this would typically be executed by a subroutine of the graphics processor 1.

As explained before also the adaptor then uses a low level hardware logic based sub-system and also a graphics processor for allowing the generation of a multi-window display wherein both high level and low level applications can execute simultaneously.

By allocating separate memory parts to the high level graphics processor based tasks and to the low level hardware logic based sub-system, both can execute concurrently in the same overall system.

The card shown in Fig. 2 is a next generation, ISA compatible display adapter for personal computers. Based on a hybrid combination of a TMS34010 Graphics System Processor (GSP) and a custom designed hardware support chip it is compatible with existing register based display adapter standards such as Video Graphics Array (VGA) and also software based display standards such as Texas Instruments Graphics Architecture (TIGA).

Features

- IBM XT/AT compatible graphics adapter card
- TMS34010 graphics system processor based
- Short AT format
- 100% Hardware VGA register compatible
- 100% VGA BIOS compatible
- VGA pass-through option
- TIGA graphics manager and communication driver available on board
- Supports 640 by 480,800 by 600 and 1024 by 768 resolutions.
- Compatible with all standard IBM PS/2 and multisync monitors
- Compatible with fixed frequency monitors in VGA and TIGA modes
- Interlaced and non-interlaced output in 1024 x 768 mode
- Modular memory design with 1M VRAM and from 512K to 2M DRAM

Memory size

The basic card is populated with 1 Mbyte of VRAM 10 and 512 Kbytes of DRAM 11. This is adequate memory for all display modes up to and including 1024 by 768 in 256 colours. It also provides enough memory for simultaneous operation in both TIGA and VGA modes with distinct frame buffers and also provides working storage and memory space for downloaded extensions to TIGA such as when using MS-Windows. For operational modes which require even greater amounts of memory, such as X windows, for example, a factory expansion option is available to increase the amount of DRAM to 2 Mbytes. The VRAM size remains 1Mbyte and display resolutions are the same.

Hardware description

General

As explained before the graphics adapter according to the invention is based on the Texas Instruments TMS34010 Graphics System Processor (GSP12). The latter provides all the intelligence and horsepower for high speed advanced graphics manipulation, whilst an associated ASIC device working in conjunction with the GSP provides the registers and hard-wired logic functions necessary to achieve full hardware IBM VGA compatibility.

PC bus interface

The PC bus interface is compatible with both 8 and 16 bit ISA standard system buses. In addition it will automatically self configure to the relevant 8 or 16 bit mode depending upon the host used.

Bus operation is specified within the range 4.77 MHz up to 10 MHz.

PC memory and I/O mapping

The present invention allows the card to appear to the PC hardware as essentially two independent

adapters on the same physical card. This is particulary so for the address decoding and mapping into PC memory and I/O space as shown in Fig. 3.

The card permits the mapping of three essentially independent functions into the host systems memory and I/O space : VGA display adapter registers and frame buffers, VGA BIOS memory and the Graphics Processor interface registers.

The locations of the first two of these functions are fixed and compatible with industry standard VGA practice as shown in the table below.

VGA standard function	I/O address	Memory address
BIOS firmware routines		C000:0000-C000:7FEF
Fixed registers	3CO-3CF	
Monochrome registers	3BO-3BF	
Colour resgisters	3DO-3DF	
Monochrome text buffer		B000:0000-B000:7FFF
Colour text buffer		B800:0000-B800:7FFF
Colour graphics buffer		A000:0000-A000:FFFF
Extended graphics buffer		A000:0000-B000:FFFF

Memory and I/O locations of standard VGA display functions

The third function consists of the TMS34010 Graphics System Processor host interface registers used for high level command communication and software interfaces such as TIGA. So as to interfere as little as possible with the standard memory and I/O use of a typical PC this interface can be either memory mapped to an unused part of the VGA BIOS memory space or I/0 mapped to one of two user selectable as shown in the table below. Thus a separate or direct input or output of display data is provided, for example by means of memory mapping as described.

GSP host register	Memory mapped option	I/O mapped option 1	I/0 mapped option 2
HSTDATA msb	C000:7FF8	0294	0284
lsb	COOO:7FF9	0295	0285
HSTCTL msb	COOO:7FFA	0296	0286
lsb	C000:7FFB	0297	0287
HSTADRL msb	COOO:7FFC	0290	0280
lsb	COOO:7FFD	0291	0281
HSTADHR msb	COOO:7FFE	0292	0282
lsb	COOO:7FFF	0293	0283

Optional GSP host register interface mapping

Additionally, the on board BIOS PROMS may be disabled as a user option to allow for circumstances such as the use of the card in a PC which already contains a traditional VGA adapter. This is known as 'pass-through' mode and is described below.

For operational modes associated with VGA compatibility the display adapter is accessed in an identical manner to standard VGA practice.

BIOS extension

A pair of on-board EPROMS's 13 contains the system BIOS exension program for compatible operation. These will permit maximum speed 16 bit operation in applicable machines and will also allow 8 bit operation in machines with this bus size.

The EPROM's also contain the GSP support program necessary for full board operation. This is transferred from the BIOS EPROM to GSP RAM by the PC boot procedure.

The EPROM's each contain a maximum of 32k bytes.

Graphics System Processor (12)

The display adapter uses the Texas Instruments TMS34010 Graphics System Processor (GSP) for high performance, flexibility and ease of customisation. This is a 32 specialised graphics microprocessor with high speed RISC type pipelined architecture capable of execution speeds up to 7,5 million instructions per second. The instruction set is both general purpose allowing full development and execution of software written in high level languages such as C, and specialised allowing software efficiency and performance when manipulating graphics data.

VGA interface chip (14)

The hardware compatibility with VGA standard is obtained through the dedicated VGA interface chip. In will be noted that this device does not contain a fully independent VGA sub-system, but rather, provides those hardware features required for full 100% 'register level' VGA. These features include control registers, real-time logic functions and specific address and data mapping as will be described. In accordance with a feature of the present invention full VGA functionality is provided by the GSP.

The VGA interface device also provides the address and data decoding for both the PC host bus and the local memory system bus.

VGA Pass-through Option

In addition to supplying all the necessary elements for full VGA compatible operation, the card is also capable of operating with another VGA card whilst still needing only one monitor. In this mode, called VGA pass-through mode, the card generates the TIGA compatible display portion and the other VGA card the VGA display portion. The latter is routed from the pure VGA card via a pass-through cable to the feature connector of the card whence it is fed the local palette input and then to the single monitor output. In this mode logic on the card ensures that its local palette contains a copy of the original VGA palette and that the on-board BIOS PROMS and VGA input registers are disabled.

Local Memory

Local memory is the term used to refer to memory contained on the display adapter which is used by the GSP and/or VGA support device but which is not accessible directly from the PC address and data buses or PC application software.

The display adapter is intrinsically modular in terms of memory size. Theoretically any size of memory could be used depending upon the display resolution and number of colours required and the amount of local software. The memory is composed of a mixture of VRAM which is used for display purposes and DRAM which is used for non-display purposes such as program, character generators etcetera. The baseline system contains 1M byte of VRAM and 512K of DRAM. The extended system contains 1M byte of VRAM and 2M bytes of DRAM.

Colour Palette (16)

The display adapter contains an industry standard VGA compatible palette with a maximum pixel frequency of 65MHz connected at the output of a TI34098 CRT control chip 15. This allows display resolutions up to 1024 by 768 in 256 colours at a frame frequency of 60Hz. The electrical characteristics and drive capability of the monitor ouput are compatible with standard fixed mode and multi-sync type monitors.

The colour of the overscan border is controlled through software via programmable on-board registers. The width and height of the border can be programmed independently from each other and other paramaters.

The polarity of both the horizontal and vertical synchronisation signals set to the attached monitor can be individually controlled by GSP software.

The logical states of pins 9, 10 and 11 of the monitor connector can be determined by GSP software in order to allow for automatic monitor type detection where applicable.

The pixel output frequency can be selected by GSP software between at least 4 non-harmonically related frequencies which all lie in the range 5-65MHz. In addition, some sub-harmonics of these four frequencies are also available and selectable by GSP software. Under normal circumstances the board will

be equipped with frequencies allowing compatibility with standard VGA operating modes and monitors.

Due to the unique architecture of the card, the actual display function is entirely independent of the VGA sub-system and is entirely under GSP software control. This gives a very important increase in system flexibility since display output can be customised to individual user needs such as flat pannel displays or even fixed frequency monitors even through several display resolutions are used.

Logic based hardware subsystem (14)

The logic based hardware subsystem is a single device containing a hardware VGA subsystem designed to operate in conjuction with a TMS34010 Graphics System Processor (GSP).

The logic based hardware subsystem allows the system designer to create a single board level system for the PC environment with both high performance graphics compatibility through TIGA and backward hardware compatibility at the VGA register and BIOS levels. The logic based hardware subsystem provides those essential hardware elements of the VGA standard not already provided by the GSP system such as I/O registers and real-time logic functions such as rotate, mask, etcetera.

In addition, and most important, because the logic based hardware subsystem is essentially providing autonomous VGA hardware support to the TMS34010, both systems can operate simultaneously and independantly from each other, using either separate or shared memory areas in the common local memory. This enables free mixing of 'hardware' generated VGA displays with 'software' generated displays from another environment such as a windowing environment program running under TIGA. Because the VGA 'window' is hardware generated, no performance trade-off due to emulation need be tollerated in any mode. Furthermore, because of the logic based hardware subsystem's essential dissociation of PC side accesses to the VGA hardware model and the display of the resultant memory areas, any host machine capable of multi-tasking in virtual address spaces can have multiple active hardware VGA windows on the same physical display at the same time.

For optimum integration and with a view to reducing the total number of devices in the final TIGA/VGA system, the logic based hardware subsystem chip also contains the logic interfaces between the PC expansion bus and the GSP and between the GSP and the shared memory system.

The functions contained in the logic based hardware subsystem 14 are shown in Figure 4.

The block diagram of the internal architecture of the logic based hardware subsystem appears in Figure 5.

This logic based hardware subsystem comprises the following elements :

- a PC bus interface 20
- a Sequence controller 21
- an Address mapper 22
- a Data mapper 23
- Internal registers 24
- a Display Controller 25
- a GSP/LAD bus interface 26
- an Arbitration controller 27
- a Memory controller 28

The PC bus interface 20 receives as inputs the PC control, address and DATA signals and provides corresponding signals to the sequence controller 21, the address mapper 22 and the DATA mapper 23 respectively.

The output of the sequence controller 21 is connected to the input of the display controller 25 whose output is connected to an input of the GSP/LAD interface 26. Another input of this interface 26 is connected to the LAD bus and an output of this interface 26 is connected to an input of the memory controller 28.

The ouput of the address mapper 22 is connected to an input of the internal registers 24 and to an input of the access arbitration controller 27. Another input of this controller 27 is connected to an output of the GSP/LAD BUS interface 26 and an ouput of the arbitration controller is connected to an input of the memory controller 28.

The ouput of the data mapper 25 is connected to another input of the internal registers 24 and to another input of the access arbitration controller 27. Another output of this controller 27 is connected to the memory controller 28.

The output of this memory controller 28 is connected to the corresponding second part of the video memory.

As explained before this sub-system provides autonomous VGA hardware support to the TMS34010.

As explained before this sub-system provides autonomous VGA hardware support to the TMS34010.

Further details concerning the display adapter and in particular the logic hardware subsystem are disclosed in the following text.

Contents

1 Introduction

1.1 General concept

The TMS34096 is a single device containing a hardware VGA subsystem designed to operate in conjuction with a TMS34010 Graphics System Processor (GSP).

The TMS34096 allows the system designer to create a single board level system for the PC environment with both high performace graphics compatibility through *TIGA* and backward hardware compatibility at the VGA register and BIOS levels. The TMS34096 provides those essential hardware elements of the VGA standard not already provided by the GSP system such as i/o registers and real-time logic functions such as rotate, mask, etcetera.

In addition, and most important, because the TMS34096 is essentially providing autonomous VGA hardware support to the TMS34010, both systems can operate simultaneously and independantly from each other, using either seperate or shared memory areas in the common local memory. This enables free mixing of 'hardware' generated VGA displays with 'software' generated displays from another environment such as a windowing environment program running under *TIGA*. Because the VGA 'window' is hardware generated, no performance trade-off due to emulation need be tollerated in any mode. Furthermore, because of the 34096's essential dissassociation of PC side accesses to the VGA hardware model and the display of the resultant memory areas, any host machine capable of multi-tasking in virtual address spaces can have multiple active hardware VGA windows on the same physical display at the same time.

For optimum integration and with a view to reducing the total number of devices in the final *TIGA*/VGA system, the TMS34096 also contains the logic interfaces between the PC expansion bus and the GSP and between the GSP and the shared memory system.

The functions contained in the TMS34096 are shown in Figure 1.1.

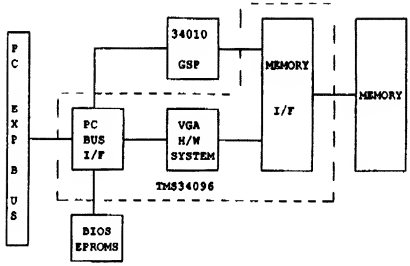


Figure 1.1: TMS34096 Integrated Functions

1.2 System Architecture

A block diagram of a typical board level system based upon the TMS34096 VGA Window Chip appears in Figure 1.2

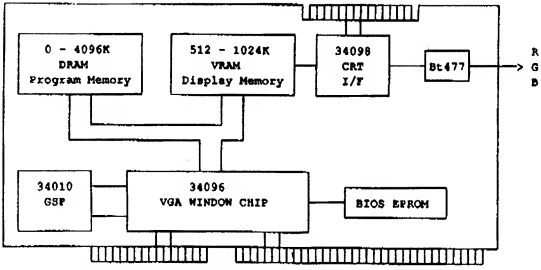


Figure 1.2: Board Level System based upon the TMS34096

The board level system shown in Figure 1.2 is composed of seven main elements:

- *TMS34010 Graphics System Processor*
- *TMS34096 VGA Window Chip*
- *VGA extension BIOS in EPROM*
- *VRAM memory for display storage*
- *DRAM memory for program and working storage*
- *T134098 CRT control chip*
- *Colour palette*

Detailed descriptions of these devices can be found in their respective specifications, and the way in which they inter-operate to perform the complete display adapter function is discussed in the *TIGA-CARD Technical Reference Manual*. A list of all related documentation can be found in the bibliography section at the end of this specification.

2 Device Architecture

2.1 Block Diagram

The block diagram of the internal architecture of the TMS34096 appears in Figure 2.1.

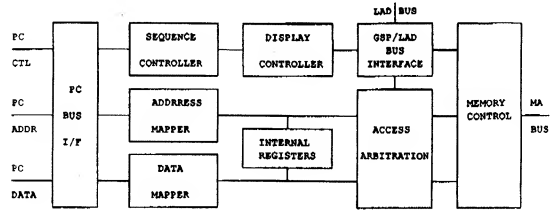


Figure 2.1: TMS34096 Internal Architecture

The block diagram of the TMS34096 shown in Figure 2.1 is composed of the following elements:

- PC bus interface
- Sequence controller
- Address mapper
- Data mapper
- Internal registers
- GSP/LAD bus interface
- Arbitration control
- Memory controller

3 Device description

3.1 PC Interface

3.1.1 General

The PC interface provides the necessary address decoding and control signal logic to interface the internal VGA subsystem, associated BIOS PROMs and the companion TMS34010 Graphics System Processor to the host PC bus. It is compatible with 8 and 16 bit ISA busses and automatically configures to the correct bus width depending upon the PC slot used.

In some circumstances it may be desirable to disable 16 bit operation such as when another board in the system occupies the same 128K address range. This can be effected via an external jumper.

The three sections of the interface are essentially independant and are described in the following paragraphs.

3.1.2 TMS34010 Graphics System Processor Interface

The TMS34096 provides the necessary logic to connect the host interface of the TMS34010 GSP to the PC bus. Details of the recomended connections between the TMS34096 and the TMS34010 GSP can be

found in Appendix I.

The GSP can be either memory or I/O mapped in the PC address space with two address options in the I/O space. The addresses of the various GSP host registers for each option are shown in Table 3.1.

GSP Register	Byte Access	Memory Mapped	I/O Option 1	I/O Option 2
HSTDATA	Lsb	0C7FF8h	0294	0284
HSTDATA	Msb	0C7FF9h	0295	0285
HSTCTL	Lsb	0C7FFAh	0296	0286
HSTCTL	Msb	0C7FFBh	0297	0287
HSTADRL	Lsb	0C7FFCh	0290	0280
HSTADRL	Msb	0C7FFDh	0291	0281
HSTADRH	Lsb	0C7FFEh	0292	0282
HSTADRH	Msb	0C7FFFh	0293	0283

The three options are selected via two input pins IO1 and IO2 as shown in Table 3.2. Both inputs have internal pull-ups allowing control via two simple SPST switches or straps as described in Appendix I.

GHM1	GHM0	GSP Host Mapping
1	1	Memory mapped
1	0	I/O option 1
0	X	I/O option 2

Table 3.2: GSP host interface mapping selection

The HRDY and HINT lines from the GSP are combined with ready and interrupt lines from the VGA sub-system to generate the IOCHRDY and PCINT signals sent to the host PC bus.

3.1.3 VGA Sub-system Interface

The Mapping of the VGA sub-system into PC space is fixed and defined by industry standard VGA practice. Various parts of the VGA subsystem can be enabled or disabled in order to provide compatibility with preceding display standards such as MDA, CGA and EGA. These options are controlled by on-chip I/O registers and are described in section 3.2. The various mapping options are shown in Table 3.3.

VGA standard function	I/O address	Memory address
Fixed registers	3C0 - 3CF	
Monochrome registers	3B0 - 3BF	
Colour registers	3D0 - 3DF	
Monochrome text buffer		B000:0000 - B000:7FFF
Colour text buffer		B800:0000 - B800:7FFF
Colour graphics buffer		A000:0000 - A000:FFFF
Extended graphics buffer		A000:0000 - B000:FFFF

Table 3.3: Mapping of VGA sub-system into PC I/O and memory space

3.1.4 BIOS Extension Mapping

The TMS34096 provides decoded enable signals for a pair of external PROMs containing the VGA BIOS extension. These are mapped into PC memory address space as shown below:

BIOS PROM memory mapping: C0000-C7FEF

BIOS PROM decoding can be enabled or disabled via the state of the BIOS enable input pin ENBIOS. This pin when driven high or left unconnected enables the BIOS decoding and when pulled low disables it. This enables BIOS decoding to be controlled via a simple SPST switch or jumper as described in Appendix I.

3.1.5 Data Bus Buffer Control

The TMS34096 generates signals to control external bi-directional bus transceivers. The direction and enable control for these transceivers are generated by a combination of the logic outputs from each of the sections described above. Additionally, these signals allow byte-swapping from lsb to msb for compatibility with 8 bit PC slots. The recommended implementation of the external bus buffering is described in Appendix I.

3.2 VGA I/O registers

3.2.1 Address decoding

The VGA Window chip decodes an area of I/O space from 03B0 to 03DF. This area is used to access two groups of registers, one which is fixed and is always located in the range 03C0 to 03CF and the other which can be located either in the range 03B0 to 03BF or 03D0 to 03DF. Some registers are read/write, some are read-only or write-only and some have different read and write addresses. The registers are listed in Table 3.4.

Register	R/W	Address
Attribute controller index register	R/W	3C0
Attribute controller data register	W	3C0
Attribute controller data register	R	3C1
Miscelaneous output register	W	3C2
Video subsystem enable	R/W	3C3
Sequencer address register	R/W	3C4
Sequencer data register	R/W	3C5
Palette mask register	R/W	3C6
Palette address register - read mode	W	3C7
Palette state register	R	3C7
Palette address register - write mode	R/W	3C8
Palette data register	R/W	3C9
Miscelaneous output register	R	3CC
Graphics controler index register	R/W	3CE
Graphics controler data register	R/W	3CF
CRT controller index register	R/W	3X4*
CRT controler data register	R/W	3X5
Input status register 1	R	3XA
Feature control register	W	3XA

*X = B or D depending upon bit 0 of miscelaneous output register

Table 3.4: VGA register addresses in PC I/O map

The VGA registers are divided up into six groups: general registers, attribute registers, CRT controller registers, sequencer registers, graphics registers and palette registers. These registers are either accessed directly or via an indirect index/data register pair. The various groups are described in paragraphs 3.2.2 through 3.2.7.

Note that the implementation of a full VGA display adaptor requires more functionality than described in the following paragraphs; only the register functions actually implemented by the hardware of the VGA Window chip are described. The remaining register functions are performed by routines of the VGA support

program running on the associated TMS34010 Graphics System Processor.

3.2.2 General registers

Table 3.5 shows the general registers

Register name	Read	Write	GSP address	On-chip
Miscelaneous Output Register	3CC	3C2	EIBI4020	Yes
Input Status Register 0	3C2	-	EIBI4220	Yes
Input Status Register 1	3XA1	-	EIBI43A0	Yes
Feature Control Register	3CA	3XA*	EIBI41A0	No
VGA Enable Register	3C3	3C3	EIBI4030	Yes

*X = B or D depending upon bit 0 of Miscelaneous Output Register

Table 3.5: General registers

3.2.2.1 Miscelaneous Output Register

The Miscelaneous Output Register can be written at address 3C2 and read from address 3CC. Figure 3.11 shows the bits implemented in the register and Table 3.6 describes their function.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	ni	ni	ni	IAS

Figure 3.11: Miscelaneous Output Register

Bit	Symbol	Function
0	IAS	This bit specifies whether the moveable registers are decoded at addresses 3BX or 3DX. A logical zero selects 3BX.
1 - 7	ni	These bits are not implemented on chip.

Table 3.6: Miscelaneous Output Register

3.2.2.2 Input Status Register 0

Input Status Register 0 can be read from address 3C2. Figure 3.12 shows the bits implemented in the register and Table 3.7 describes their function.

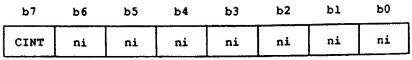


Figure 3.12: Input Status Register 0

Bit	Symbol	Function
0 - 6	ni	These bits are not implemented on chip.
7	CINT	CRT interrupt. This bit when set indicates that a CRT interrupt is pending.

Table 3.7: Input Status Register 0

3.2.2.3 Input Status Register 1

Input Status Register 1 can be read from address 3BA or 3DA.
Figure 3.13 shows the bits implemented in the register and Table 3.8 describes their function.

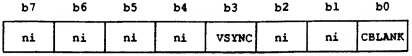


Figure 3.13: Input Status Register 1

Bit	Symbol	Function
0	CBLANK	Composite blanking signal. This bit when set indicates that the display is in either a horizontal or vertical blanking interval.
1 & 2	ni	These bits are not implemented on chip.
3	VSYNC	Vertical sync. This bit when set indicates that the display is in a vertical retrace interval.
4 - 7	ni	These bits are not implemented on chip.

Table 3.8: Input Status Register 1

3.2.2.4 VGA Enable Register

The VGA Enable Register when enabled is read and written at address 3C3. The VGA Enable Register enable control can be found in the configuration registers and is described in section 3.8.

Figure 3.14 shows the bits implemented in the register and Table 3.9 describes their function.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	ni	ni	ni	VEN

Figure 3.14: VGA Enable Register

Bit	Symbol	Function
0	VEN	VGA Enable. This bit when set enables PC accesses to the VGA I/O and memory address space.
1 - 7	ni	These bits are not implemented on chip.

Table 3.9: VGA Enable register

3.2.3 Attribute registers

Attribute registers are accessed indirectly via an index/data pair. To access any particular register, the index for that register is first written to bits b0-b4 of the index register, and the data is then written to, or read from a common data register. One particularity of the attribute registers is the addresses of the index and data registers: writing the index register and the data register are both at address 3C0. An internal flip-flop directs alternate accesses to alternate registers. The flip-flop can be reset to a know state by reading Input Status Register 1. The next access to 3C0 will then be directed to the index register.

Attribute registers can be read by writing their index value to the index register and then reading the data register from address 3C1.

Apart from the necessary logic to map PC addresses into GSP memory space as shown in Table 3.10 below, none of the Attribute Registers are implemented on-chip and their description is outside the scope of this document.

Table 3.10 lists the attribute registers and their index values.

Register name	Index	GSP address	On chip
Index register	-	E_TB_T4000	Yes
EGA palette registers	00-0F	E_TB_T4400-E_TB_T44F0	No
Attribute Mode Control	10	E_TB_T4500	No
Border Colour Register	11	E_TB_T4510	No
Colour Plane Enable Register	12	E_TB_T4520	No
Horizontal Panning Register	13	E_TB_T4530	No
Colour Select Register	14	E_TB_T4540	No
----- undefined ------	15-1F	E_TB_T4550-E_TB_T45F0	No

Table 3.10: Attribute Registers

3.2.4 CRT Controller Registers

CRT Controller registers are accessed via an indirect index/data register pair. To access any particular register, the index for that register is first written to bits b0-b4 of the index register, and the data is then written to, or read from a common data register. The addresses of the index and data registers are 3X4 and 3X5 respectively where X is either B or D depending upon the state of Miscelaneous Output Register bit 0.

Table 3.11 lists the names of the CRT Controller registers, their location in GSP memory and whether they are implemented on-chip.

Register name	Index	GSP address	On chip
Index Register	-	EIBI4140	Yes
Horizontal Total	00	EIBI4C00	No
Horizontal Display Enable End	01	EIBI4C10	No
Start Horizontal Blanking	02	EIBI4C20	No
End Horizontal Blanking	03	EIBI4C30	No
Start Horizontal Retrace	04	EIBI4C40	No
End Horizontal Retrace	05	EIBI4C50	No
Vertical Total	06	EIBI4C60	No
Overflow	07	EIBI4C70	No
Preset Row Scan	08	EIBI4C80	No
Maximum Scan Line	09	EIBI4C90	No
Cursor Start	0A	EIBI4CA0	No
Cursor End	0B	EIBI4CB0	No
Start Address High	0C	EIBI4CC0	No
Start Address Low	0D	EIBI4CD0	No
Cursor Location High	0E	EIBI4CE0	No
Cursor Location Low	0F	EIBI4CF0	No
Vertical Retrace Start	10	EIBI4D00	No
Vertical Retrace End	11	EIBI4D10	Yes
Vertical Display Enable End	12	EIBI4D20	No
Offset	13	EIBI4D30	No
Underline Location	14	EIBI4D40	No
Start Vertical Blank	15	EIBI4D50	No
End Vertical Blank	16	EIBI4D60	No
CRTC Mode Control	17	EIBI4D70	No
Line Compare	18	EIBI4D80	No
----- undefined ------	19-1F	EIBI4D90-EIBI4DF0	No

Table 3.11: CRT Controler Registers

3.2.4.1 Vertical Retrace End Register

The Vertical Retrace End Register is accessed at address 3B5 or 3D5 when the CRT Controller Index Register contains a value of 11.

Three bits of the Vertical Retrace End register are implemeted on-chip as shown in Figure 3.5. Their function is described in Table 3.12.

b7	b6	b5	b4	b3	b2	b1	b0
P0-7	ni	EVI	CVI	ni	ni	ni	ni

Figure 3.5: Vertical Retrace End Register

Bit	Symbol	Function
0 - 3	ni	These bits are not implemented on chip.
4	CVI	Clear Vertical Interrupt. This bit when set to zero clears a vertical retrace interrupt. It must be set to one to allow interrupts to be generated.
5	EVI	Enable Vertical Interrupt. This bit when set to zero enables a vertical interrupt condition to appear on the external INTOUT pin.
6	ni	This bits is not implemented on chip.
7	PO-7	Protect Registers 0 - 7. This bit when set to one disables writing to CRT controller registers 0 to 7.

Table 3.12: Vertical Retrace End Register

3.2.5 Sequencer Registers

Sequencer registers are accessed via an indirect index/data register pair. To access any particular register, the index for that register is first written to bits b0-b4 of the index register, and the data is then written to, or read from a common data register. The addresses of the index and data registers are 3C4 and 3C5 respectively.

Table 3.13 lists the names of the Sequencer registers, their location in GSP memory and whether they are implemented on-chip.

Register name	Index	GSP address	On chip
Index Register	-	E_IB_I4040	Yes
Reset	00	E_IB_I4800	No
Clocking Mode	01	E_IB_I4810	No
Map Mask	02	E_IB_I4820	Yes
Character Map Select	03	E_IB_I4830	No
Memory Mode	04	E_IB_I4840	Yes
----- undefined ------	05-1F	E_IB_I4850-E_IB_I49F0	No

Table 3.13: Sequencer Registers

3.2.5.1 Map Mask Register

The Map Mask Register is accessed at address 3C5 when the Sequencer Index Register contains a value of 02.

Four bits of the Map Mask Register are implemeted on-chip as shown in Figure 3.6. Their function is described in Table 3.14.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	ME3	ME2	ME1	ME0

Figure 3.6: Map Mask Register

Bit	Symbol	Function
0 - 3	ME0-3	Map Enable 0 - 3. These bits control write accesses the the frame buffer.
4 - 7	ni	These bits are not implemented on chip.

Table 3.14: Map Mask Register

3.2.5.2 Memory Mode Register

The Memory Mode Register is accessed at address 3C5 when the Sequencer Index Register contains a value of 04.

Two bits of the Memory Mode Register are implemeted on-chip as shown in Figure 3.7. Their function is described in Table 3.15.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	Chain4	Odd/even	ni	ni

Figure 3.7: Memory Mode Register

Bit	Symbol	Function
0 - 1	ni	These bits are not implemented on chip.
2	Odd/$\overline{\text{even}}$	Odd/even address mapping. This bit controls the mapping of display memory into the host processors address map. It is set to zero for all text modes and modes 4 & 5 and to one for all other graphics modes.
3	Chain4	Chained address mapping. This bit controls the data and address mapping from host processor to local memory space. It is set to 1 for mode 13 and to zero for all other modes.
4 - 7	ni	These bits are not implemented on chip.

Table 3.15: Memory Mode Register

3.2.6 Graphics Controller Registers

Graphics registers are accessed via an indirect index/data register pair. To access any particular register, the index for that register is first written to bits b0-b4 of the index register, and the data is then written to, or read from a common data register. The addresses of the index and data registers are 3CE and 3CF respectively.

Table 3.16 lists the names of the Graphics registers, their location in GSP memory and whether they are implemented on-chip.

Register name	Index	GSP address	On chip
Graphics Index Register	-	E_TB_T40E0	Yes
Set/Reset	00	E_TB_T4A00	Yes
Enable Set/Reset	01	E_TB_T4A10	Yes
Colour Compare	02	E_TB_T4A20	Yes
Data Rotate	03	E_TB_T4A30	Yes
Read Map select	04	E_TB_T4A40	Yes
Graphics Mode	05	E_TB_T4A50	Yes
Miscellaneous	06	E_TB_T4A60	Yes
Colour Don't Care	07	E_TB_T4A70	Yes
Bit Mask	08	E_TB_T4A80	Yes
----- undefined ------	09-1F	E_TB_T4A90-E_TB_T4BF0	No

Table 3.16: Graphics Registers

3.2.6.1 Set/Reset Register

The Set/Reset Register is accessed at address 3CF when the Graphics Index Register contains a value of 00.

Four bits of the Set/Reset Register are implemeted on-chip as shown in Figure 3.8. Their function is

described in Table 3.17.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	SR3	SR2	SR1	SR0

Figure 3.8: Set/Reset Register

Bit	Symbol	Function
0 - 3	SR0 - 3	Set/Reset bits 0 - 3. These bits are the values written to planes 0 to 3 in write mode 0 (cf para. 3.5.3.2) when set/reset mode is enabled. Each bit can be enabled individually via the Set/Reset Enable Register (cf para. 3.2.6.2)
4 - 7	ni	These bits are not implemented on chip.

Table 3.17: Set/Reset Register

3.2.6.2 Set/Reset Enable Register

The Set/Reset Register is accessed at address 3CF when the Graphics Index Register contains a value of 01.

Four bits of the Set/Reset Enable Register are implemeted on-chip as shown in Figure 3.9. Their function is described in Table 3.18.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	SRE3	SRE2	SRE1	SRE0

Figure 3.9: Set/Reset Enable Register

Bit	Symbol	Function
0 - 3	SRE0 - 3	Set/Reset Enable bits 0 - 3. These bits enable individual planes to be written with the corresponding value in the Set/Reset Register (cf para 3.2.6.1) when in Write Mode 0 (cf para 3.5.3.3) and Set/Reset mode is enabled
4 - 7	ni	These bits are not implemented on chip.

Table 3.18: Set/Reset Enable Register

3.2.6.3 Colour Compare Register

The Colour Compare Register is accessed at address 3CF when the Graphics Index Register contains a value of 02.

Four bits of the Colour Compare Register are implemeted on-chip as shown in Figure 3.10. Their function is described in Table 3.19.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	CC3	CC2	CC1	CC0

Figure 3.10: Colour Compare Register

Bit	Symbol	Function
0 - 3	CC0 - 3	Colour Compare bits 0 - 3 These bits represent the value used to compare with the pixel value read from the display memory when in Read Mode 1. (cf para 3.2.6.6). Individual bits may be ignored depending upon the value set in Colour Don't Care Register (cf para. 3.2.6.8)
4 - 7	ni	These bits are not implemented on chip.

Table 3.19: Colour Compare Register

3.2.6.4 Data Rotate Register

The Data Rotate Register is accessed at address 3CF when the Graphics Index Register contains a

value of 03.

Five bits of the Data Rotate Register are implemeted on-chip as shown in Figure 3.11. Their function is described in Table 3.20.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	FS1	FS0	RC2	RC1	RC0

Figure 3.11: Data Rotate Register

Bit	Symbol	Function
0 - 2	RC0 - 2	Rotate Count bits 0 - 2 These bits represent a binary encoded value of the number of bit positions by which PC data must be rotated to the right when writing to memory in Write Mode 0 (cf para 8.8)
3 - 4	FS0 - 1	Function Select 0 - 1 These bits control the logical function to be applied between PC data being written and data in the read-back latches (cf para 3.5.3.4): FS1 FS0 Function 0 0 Replace 0 1 AND 1 0 OR 1 1 XOR
5 - 7	ni	These bits are not implemented on chip.

Table 3.20: Data Rotate Register

3.2.6.5 Read Map Select Register

The Read Map Select Register is accessed at address 3CF when the Graphics Index Register contains a value of 04.

Two bits of the Read Map Select Register are implemeted on-chip as shown in Figure 3.12. Their function is described in Table 3.21.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	ni	ni	RMS1	RMS0

Figure 3.12: Read Map Select Register

Bit	Symbol	Function
0 - 1	RMS0 - 1	Read Map Select bits 0 and 1 These bits represent a binary encoded value of the plane to be read in Read Mode 0 (cf para 3.2.6.6)
2 - 7	ni	These bits are not implemented on chip.

Table 3.21: Read Map Select Register

3.2.6.6 Graphics Mode Register

The Graphics Mode Register is accessed at address 3CF when the Graphics Index Register contains a value of 05.

Three bits of the Graphics Mode Register are implemeted on-chip as shown in Figure 3.13. Their function is described in Table 3.22.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	RM	ni	WM1	WM0

Figure 3.13: Graphics Mode Register

Bit	Symbol	Function
0 - 1	WM0 - 1	Write Mode bits 0 and 1 These bits represent a binary encoded value of the write mode (cf para 3.5.2.3) WM1 WM0 Write mode 0 0 0 - Normal write 0 1 1 - Latch write 1 0 2 - Packed write 1 1 invalid
2	ni	This bit is not implemented on chip
3	RM	Read mode When set, this bit selects Colour Compare Mode (cf para #.#)
4 - 7	ni	These bits are not implemented on chip.

Table 3.22: Graphics Mode Register

3.2.6.7 Graphics Miscelaneous Register

The Graphics Miscelaneous Register is accessed at address 3CF when the Graphics Index Register contains a value of 06.

Three bits of the Graphics Miscelaneous Register are implemeted on-chip as shown in Figure 3.14. Their function is described in Table 3.23.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	MM1	MM0	ni	GRM

Figure 3.14: Graphics Miscelaneous Register

Bit	Symbol	Function
0	GRM	Graphics Mode This bit when set indicates the the chip should operate in Graphics Mode.
1	ni	This bit is not implemented on chip
2 - 3	MM0 - 1	Memory Mapping bits 0 - 1 These bits control the address of the frame buffer mapped into PC address space: MM1 MM0 PC address 1 0 B0000 - B7FFF 1 1 B8000 - BFFFF 0 1 A0000 - AFFFF 0 0 A0000 - BFFFF
4 - 7	ni	These bits are not implemented on chip.

Table 3.23: Graphics Miscelaneous Register

3.2.6.8 Colour Don't Care Register

The Colour Don't Care Register is accessed at address 3CF when the Graphics Index Register contains a value of 07.

Four bits of the Colour Don't Care Register are implemeted on-chip as shown in Figure 3.15. Their function is described in Table 3.24.

b7	b6	b5	b4	b3	b2	b1	b0
ni	ni	ni	ni	CDC3	CDC2	CDC1	CDC0

Figure 3.16: Colour Don't Care Register

Bit	Symbol	Function
0 - 3	CDC0 - 3	Colour Don't Care bits 0 - 3 These bits Are used in conjunction with the Colour Compare Register (cf para. 3.2.6) when in Read Mode 1. Each bit when set causes the value read from that plane to be a 'don't care' for the colour compare operation.
4 - 7	ni	These bits are not implemented on chip.

Table 3.24: Colour Don't Care Register

3.2.6.9 Bit Mask Register

The Bit Mask Register is accessed at address 3CF when the Graphics Index Register contains a value of 08.

All eight bits of the Bit Mask Register are implemeted on-chip as shown in Figure 3.16. Their function is described in Table 3.25.

b7	b6	b5	b4	b3	b2	b1	b0
BM7	BM6	BM5	BM4	BM3	BM2	BM1	BM0

Figure 3.16: Bit Mask Register

Bit	Symbol	Function
0 - 7	BM0 - 7	Bit Mask bits 0 - 7 These bits can be used to prevent certain bit positions from being modified during a host processor write to memory. Any bit cleared to zero causes the data written to memory to be taken from the processor latches rather than from the PC data bus. Note that the processor latches must have been previously loaded with the correct data by a read operation at the same address for the true mask operation to be made. Setting each bit to one allows processor data for that bit position to pass unimpeded to memory.

Table 3.25: Bit Mask Register

3.3 Configuration Registers

3.3.1 General

The TMS34096 contains several configuration registers which control the operation of the device. These registers are accessed via the GSP local bus interface and are mapped into GSP address space in the range F200 0000 to F200 00C0. Table 3.29 lists these registers together with their specific GSP address locations and sections 3.3.2 to 3.3.12 describe their contents.

Register Name	GSP Local Address	Register Description	Default Value
BASE1	F200 0000	Memory base addresses	FC0F
STATUS	F200 0010	Device status	
BASE2	F200 0020	Device configuration	
BASE3	F200 0030	Device configuration	
RAMCON	F200 0050	Local memory config	
ROWSTRT	F200 0060	Display row start	
TAPSTRT	F200 0070	Display tap point	
SPLIT	F200 0080	Split screen address	
LLL	F200 0090	Logical line length	
IOFLAG1	F200 00A0	I/O access flags	
IOFLAG2	F200 00B0	I/O access flags	

Table 3.26: Configuration Registers

3.3.2 BASE1 Register

The BASE1 Register is accessed at GSP address F200 0000. Figure 3.17 shows the bits implemented in this register and Table 3.27 described their use.

b15	b14	b13	b12	b11	b10	b09	b08	b07	b06	b05	b04	b03	b02	b01	b00
BI7	BI6	BI5	BI4	BI3	BI2	BI1	BI0	BG3	BG2	BG1	BG0	BT3	BT2	BT1	BT0

Figure 3.17: Configuration Register BASE1

Bit	Symbol	Reset	Function
0-3	BT0-BT3	F	Text area base address. These bits together with the ET bit of BASE2 register determine the address of the memory area used to store VGA text data.
4-7	BG0-BG3	0	Graphics area base address. These bits together with the EG bit of BASE2 register determine the address of the memory area used to store VGA graphics data.
8-15	BI0-BI7	FC	I/O register base address. These bits together with the EI bit of BASE2 register determine the address of the memory area used to store VGA I/O register data.

Table 3.27: Configuration Register BASE1

3.3.3 STATUS Register

The STATUS Register is accessed at GSP address F200 0010. Figure 3.18 shows the bits implemented in this register and Table 3.28 described their use.

b15	b14	b13	b12	b11	b10	b09	b08	b07	b06	b05	b04	b03	b02	b01	b00
BEN	ni	ni	ni	ni	ni	ni	ni	ni	ni	ni	ni	VR3	VR2	VR1	VR0

Figure 3.18: Configuration Register STATUS

Bit	Symbol	Function
0-3	VR0-VR3	Device revision code. A four bit value which uniquely identifies the revision of the silicon.
4-14	ni	These bits are not implemented on-chip
15	BEN	BIOS PROM enable status. This bit enables software to determine the state of the external BEN pin.

Table 3.28: Configuration Register STATUS

3.3.4 BASE2 Register

The BASE2 Register is accessed at GSP address F200 0020. Figure 3.19 shows the bits implemented in this register and Table 3.29 described their use.

b15	b14	b13	b12	b11	b10	b09	b08	b07	b06	b05	b04	b03	b02	b01	b00
EG	ET	EI	BP	ULK	DCD	SWE	VEE	OED	DSE	CRE	PPM	GDE	MWE	BEX	RES

Figure 3.??: Configuration Register BASE2

Bit	Symbol	Reset	Function	Section
0	RES	0	This bit is reserved for test purposes and must always be cleared to zero for normal operation.	
1	BEX	1	BIOS PROM extension control. This bit allows the selection of two address ranges for the external BIOS PROMs.	3.1.4
2	MWE	0	Mask write enable. This bit when set to one enables the sequencer to perform mask write cycles. When cleared, the sequencer will replace mask write cycles with read-modify-write cycles.	3.6.1
3	GDE	0	Graphics display enable. This bit when set enables the internal VGA graphics display controller.	
4	PPM	0	Palette protect mode. This bit when set causes VGA I/O accesses to the colour palette to be redirected to an area of local memory and not to the palette itself. GSP software can then fully control the real programmation of the palette device. See also CRE.	
5	CRE	0	Chip read enable. This bit when set enables the device to respond to host PC read cycles. When cleared the chip will only respond to write cycles. Note: if CRE and PPM are both cleared then the device is set into a special 'palette pass-through' mode where only write cycles to the palette I/O registers are effected and these are sent directly to the hardware palette device.	

Table 3.29: Configuration Register BASE2

Bit	Symbol	Reset	Function	Section
6	DSE	0	Double Scan Enable. This bit when set causes the internal graphics display controller to display each scan line twice.	
7	OED	0	Odd/Even Display. This bit when set instructs the internal graphics display controller to display even and odd scan lines from different areas of memory.	
8	VEE	0	VGA Enable Enable. This bit when set enables the VGA Enable Register described in section 3.2.2.4. When cleared it disables the register and automatically enables the VGA interface.	
9	SWE	0	Swizzle Enable. This bit, when set causes data bytes writen to and read from plane 2 of memory whilst in text mode to be mirrored end to end.	
10	DCD	0	Disable CBLANK Delay. This bit, when set disables the digital delay normally applied to the internal signal taken from the CBLANK input.	
11	ULK	0	Chip Unlock Bit. Ths bit when cleared disables the entire chip PC host VGA interface.	
12	BP	0	Palete Base Address. This bit when set in conjunction with PPM being set causes read and writes to the palette to be redirected to a memory area starting at address E1E1BI8000. If cleared the address becomes E1E1BI0000 If PPM bit is set then the state of BP has no effect.	
13	EI	1	VGA I/O register base address extension.	3.2
14	ET	1	VGA Text access base address extension	4.1.3
15	EG	0	VGA Graphics access base address extension.	4.1.6

Table 1.29: Configuration Register BASE2 (contd.)

3.3.5 BASE3 Register

The BASE3 Register is accessed at GSP address F200 0030. Figure 3.20 shows the bits implemented in this register and Table 3.30 described their use.

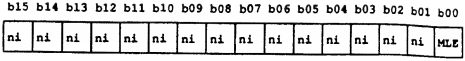


Figure 3.20: Configuration Register BASE3

Bit	Symbol	Reset	Function	Section
0	MLE	0	Midline Load Enable. This bit when set enables the internal graphics display controller to perform display update cycles during the active parts of scan lines.	3.7.?
1-15	ni		These bits are not implemented on-chip	

Table 3.30: Configuration Register BASE3

3.3.6 RAMCON Register

The RAMCON Register is accessed at GSP address F200 0050. Figure 3.21 shows the bits implemented in this register and Table 3.31 described their use.

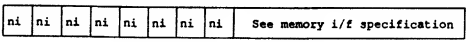


Figure 3.21: Configuration Register RAMCON

Bit	Symbol	Reset	Function
0-7			These bits control the memory configuration. For further details refer to the TMS34096 Local Memory Interface Internal Block Description
8-15	ni	0	These bits are not implemented on-chip

Table 3.31: Configuration Register RAMCON

3.3.7 ROWSTRT Register

The ROWSTRT Register is accessed at GSP address F200 0060. Figure 3.22 shows the bits implemented in this register and Table 3.32 described their use.

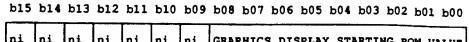


Figure 3.22: Configuration Register ROWSTRT

Bit Positions	Bit Names	Bit Description	Default Value	Related Section
b15-b09	ni	These bits are not implemented on-chip		
b08-b00	ROWSTRT	Graphics display row start value	000	

Table 3.32: Configuration Register ROWSTRT

3.3.8 TAPSTRT Register

The TAPSTRT Register is accessed at GSP address F200 0070. Figure 3.23 shows the bits implemented in this register and Table 3.33 described their use.

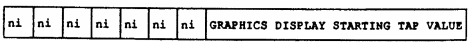


Figure 3.23: Configuration Register TAPSTRT

Bit Positions	Bit Names	Bit Description	Default Value	Related Section
b15-b09	ni	These bits are not implemented on-chip		
b08-b00	TAPSTRT	Graphics display tap start value	000	

Table 3.33: Configuration Register TAPSTRT

3.3.9 SPLIT Register

The SPLIT Register is accessed at GSP address F200 0080. Figure 3.24 shows the bits implemented in this register and Table 3.34 described their use.

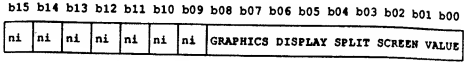


Figure 3.24: Configuration Register SPLIT

Bit Positions	Bit Names	Bit Description	Default Value	Related Section
b15-b09	ni	These bits are not implemented on-chip		
b08-b00	SPLIT	Graphics display split screen address	000	

Table 3.34: Configuration Register SPLIT

3.3.10 LLL Register

The LLL Register is accessed at GSP address F200 0090. Figure 3.25 shows the bits implemented in this register and Table 3.35 described their use.

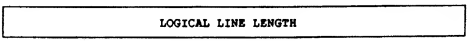


Figure 3.25: Configuration Register LLL

Bit Positions	Bit Names	Bit Description	Default Value	Related Section
b15-b00	LLL	Graphics display logical line length		

Table 3.35: Configuration Register LLL

3.3.11 IOFLAG1 Register

The IOFLAG1 Register is accessed at GSP address F200 00A0. Figure 3.26 shows the bits implemented in this register and Table 3.36 described their use.

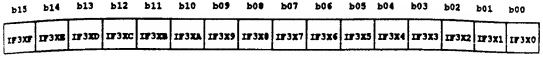


Figure 3.26: Configuration Register IOFLAG1

Bit Positions	Bit Names	Bit Description	Default Value	Related Section
b15-b00	IF3XF-IF3X0	Access flags for VGA i/o registers 3XX	0000	

Table 3.36: Configuration Register IOFLAG1

3.3.12 IOFLAG2 Register

The IOFLAG2 Register is accessed at GSP address F200 00B0. Figure 3.27 shows the bits implemented in this register and Table 3.37 described their use.

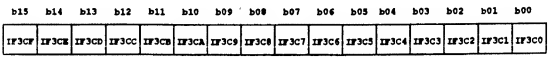


Figure 3.27: Configuration Register IOFLAG2

Bit Positions	Bit Names	Bit Description	Default Value	Related Section
b15-b00	IF3CF-IF3C0	Access flags for VGA i/o registers 3CX	0000	

Table 3.37: Configuration Register IOFLAG2

3.4 Address Mapper

3.4.1 General

The VGA address mapper is responsible for the conversion from PC side VGA logical addresses to local physical addresses. As described in paragraph 3.1.3, the TMS34096 VGA sub-system interface is mapped into PC address space in accordance with industry standard practice. The various memory areas addressed by the PC are mapped into GSP local memory space in various ways dependant upon the type of access, memory or i/o, the value of several VGA register bits and the contents of the Configuration Register base addresses as described in the following paragraphs.

3.4.2 Configuration Register Base Addresses

Two of the Configuration Registers described in section 3.3 contain base addresss information. These are BASE1 and BASE2. These two registers allow GSP software to partition the various local memory into areas used by the VGA subsystem. The areas are: I/O register block, text alpha map, text font map, graphics frame buffer and optionally palette data map. A typical implementation would also have other areas defined in the local memory space for exclusive use by the GSP and these are entirely controlled and mapped by GSP software. Figure 3.28 shows an example local memory map for a typical implementation. In addition, a typical system would be made up of a mixture of DRAM and VRAM using the former to store all non displayable areas and the latter display areas.

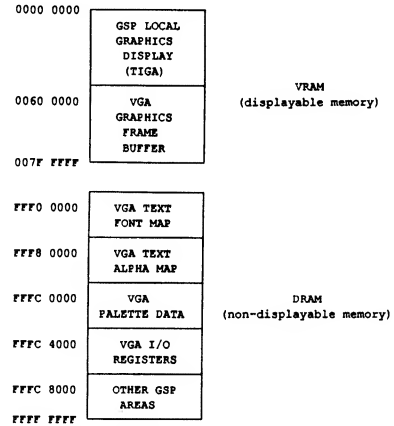


Figure 3.28:Typical Local Memory Map

3.5 VGA Datapath Logic

3.5.1 General

The TMS34096 contains logic elements in the data path between the host interface and the memory interface. This logic is active in all VGA modes of operation and conforms to industry standard VGA hardware specification. The logic includes rotation, boolean functions, bit masking, colour compare and

other functions as described in the following paragraphs.

The various logic elements described above are in general effected in a sequential manner, for example, in the write path, the first function encountered is data rotate and the last, plane masking.

3.5.2 Read Path Logic

Figure 3.29 shows the relative functional postioning of the various data path elements for the read path.

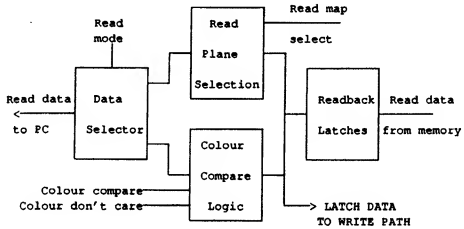


Figure 3.29: Read data path logic

3.5.2.1 Readback Latches

Some of the logic functions described rely for their operation on a set of internal registers called Readback Latches. These latches, of which there are four each containing 8 bits, latch the data read from all four logical planes whenever a host processor read cycle is performed. The data in these latches can then be used on a subsequent write cycle in conjunction with the actual processor data Bit masking, for example, will only operate correctly if the location in memory being masked has been previously read. The Readback Latches are not accessible from the PC host interface.

3.5.2.2 Read Plane Selection

The data read from logical memory and latched in the readback latches is from all four planes and is hence 32 bits wide. Since processor data is organised logically into bytes, only one byte from one plane can be read at one time. The plane to be read is selected by bits b1 and b0 of the Read Map Select Register as described in section 3.2.6.5.

In the case of a 16 bit host read, the sequencer will fetch 64 bits from memory and the 16 bit word supplied to the PC will be composed of the relevant byte from the first 32 bits and the same byte from the second 32 bits just as if the the two bytes had been read sperately and in sequence. Additionally in this case the Readback Latches will contain the 32 bit data read from the second byte.

3.5.2.3 Colour Compare Logic

The colour compare logic alows the host processor to search for a particular pixel value in the display memory. In the case of 4 bit pixels, i.e. 16 colour modes, the 32 bits of data in the Readback Latches will represent 8 pixels. Bits b0 to b3 of the Colour Compare Register described in section 3.2.6.3 are compared to the 8 pixels in turn and the corresponding bits in the byte supplied to the PC are set or cleared depending upon whether a match is found or not. Additionally, individual bit planes can be ignored during the colour compare operation by setting the corresponding bit in the Colour Don't Care Register as described in section 3.2.6.8. This process is illustrated in Figures 3.30 and 3.31, the former showing the case where all planes are compared and the latter where planes 2 and 3 are to be ignored.

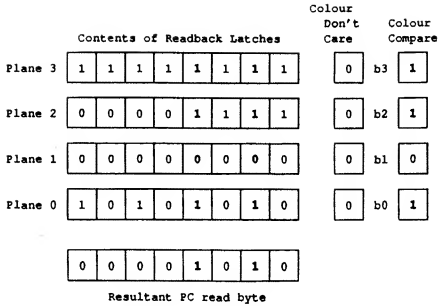


Figure 3.30: Colour Compare Operation

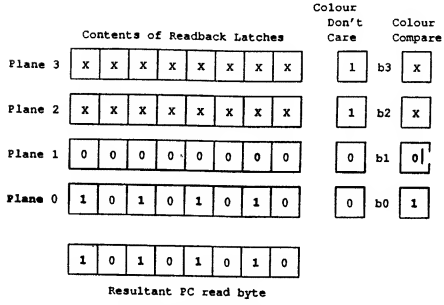


Figure 3.31: Colour Don't Care Operation

3.5.2.4 Read Mode Data Selector

During the execution of a PC read operation, the data suplied to the PC will be either the result of the colour compare operation as described above, or the data read directly from the memory plane as selected by the read map select logic. The choice is determined by the state of bit b3 of the Graphics Mode Register as described in section 3.2.6.6. If b3 of this register is set, the colour compare result will be supplied, and when cleared the direct memory data will be used.

3.5.3 Write Path Logic

Figure 3.32 shows the relative functional postioning of various data path elements for the write path.

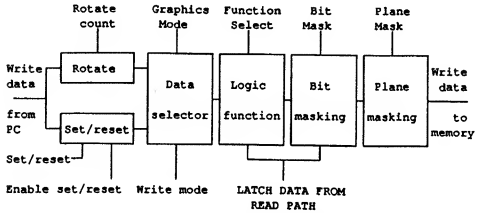


Figure 3.32: Write data path logic

3.5.3.1 Data Rotate

The data rotate logic rotates the PC data by the binary encoded value of bits b0 to b3 of the Data Rotate Register described in section 3.2.6.4. Data is rotated to the right, that is towards the least significant bit. Note that in the case of a 16 bit processor write, the two bytes will be rotated individually as would have occured if they were written seperately.

3.5.3.2 Set/Reset Logic

The set/reset logic generates 32 bits of data to be written to the memory controlled by bits b0 to b3 of the Set/Reset Register and bits b0 to b3 of the Set/Reset Enable Register described in sections 3.2.6.1 and 3.2.6.2 respectively. Bits 0 to 3 of these registers correspond to planes 0 to 3 in the memory. If the corresponding bit in the Set/Reset Enable Register is cleared the processor data is used directly for that plane. If the bit is set, however, the data bit is taken from the Set/Reset Register.

3.5.3.3 Write Mode

Bits b0 and b1 of the Graphics Mode Register described in section 3.2.6.6 control the mode of the write operations. Table 3.38 shows the three possible modes of writing data to the memory.

Graphics Mode Register b1	b0	Write Mode
0	0	Normal
0	1	Latched
1	0	Packed

Table 3.38: Write Mode Selection

In normal write mode, the processor data is written directly to memory. In Latched write mode, the contents of the Readback Latches are written to memory. In Packed write mode, the four planes are filled with the least significant four bits of the processor write data as shown in Figure 3.33

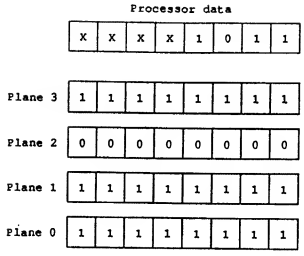


Figure 3.33: Packed Write Operation

3.5.3.4 Logic Function

Data written to memory can be combined with data from the Readback Latches in a variety of ways depending on the states of bits b3 and b4 of the Data Rotate Register described in section 3.2.6.4. Table 3.39 shows the four possible combinations of these bits.

Data Rotate Register b1	b0	Logical Function
0	0	Replace
0	1	AND
1	0	OR
1	1	XOR

Table 3.39: Logical Function Selection

3.5.3.5 Bit Masking

Individual pixels in the display memory can be protected from modification by the bit masking function. The 32 memory bits must first be stored in the Readback Latches by performing a memory read operation. The contents of the Bit Mask Register described in section 3.2.4.9 then controls whether the processor data or the contents of the Readback Latches are used to write to memory.

3.5.3.6 Plane Masking

The values of bits b0 to b3 of the Map Mask Register described in section 3.2.5.1 determine the memory planes will be up:lated during a processor write operation. If the corresponding bit is cleared then that plane will be immune to change. This function will occur independantly of the contents of the Readback Latches.

3.6 Sequencer Controller

3.6.1 General

The sequencer is responsible for generating all timing requirements for memory acesses. Due to the nature of the mapping between host data space and local data space as described in section 4, a single PC read or write cycle can require a single, double or quadruple local read, write or read-modify-write cycle corresponding to 8, 16, 32 or 64 bits of local data.

In the case of read-modify-write cycles, these may be effected as VRAM masked write cycles if the latter are enabled via the MWE bit in the BASE1 configuration register.

3.6.2 Cycles Timing

All timing is generated from the master input clock CLKIN. All memory access cycles are composed of integer numbers of CLKIN cycles.

There are three basic cycle lengths: single (16 bits), double (32 bits) and quadruple (64 bits). Each length type has in turn four different forms: read, write, read-modify-write and mask write making twelve different cycle sequences in all.

The timing of the twelve different cycles are described in the sections that follow.

3.6.2.1 Single Read Cycle

The timing of the single, 16 bit local read cycle is shown in the figure below:

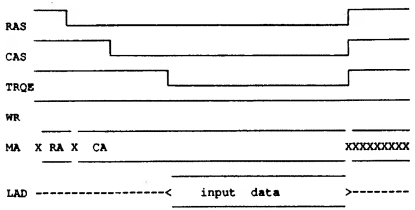


Figure 3.34: Single, 16 Bit Local Read Cycle

3.6.2.2 Single Write Cycle

The timing of the single, 16 bit local write cycle is shown in the figure below:

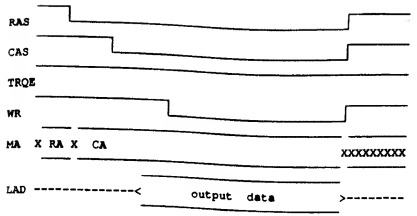


Figure 3.35: Single, 16 Bit Local Write Cycle

3.6.2.3 Single Read-modify-write Cycle

The timing of the single, 16 bit local read-modify-write cycle is shown in the figure below:

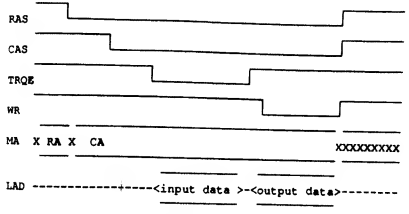


Figure 3.36: Single, 16 Bit Local Read-modify-write Cycle

3.6.2.4 Single Mask Write Cycle

The timing of the single, 16 bit local mask write cycle is shown in the figure below:

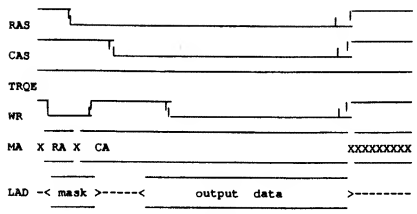


Figure 3.37: Single, 16 Bit Local Mask Write Cycle

3.6.2.5 Double Read Cycle

The timing of the double, 32 bit local read cycle is shown in the figure below:

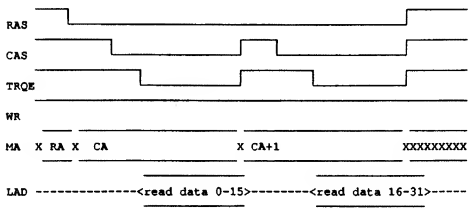


Figure 3.38: Double, 32 Bit Local Read Cycle

3.6.2.6 Double Write Cycle

The timing of the double, 32 bit local write cycle is shown in the figure below:

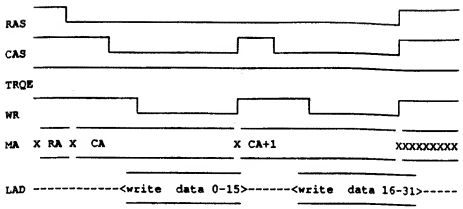


Figure 3.39: Double, 32 Bit Local Write Cycle

3.6.2.7 Double Read-modify-write Cycle

The timing of the double, 32 bit local read-modify-write cycle is shown in the figure below:

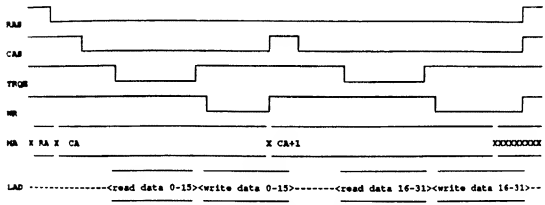


Figure 3.40: Double, 32 Bit Local Read-modify-write Cycle

3.6.2.8 Double Mask Write Cycle

The timing of the double, 32 bit local mask write cycle is shown in the figure below:

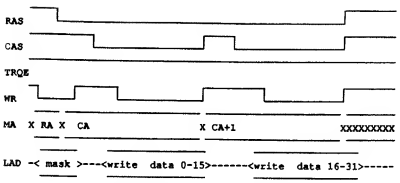


Figure 3.41: Double, 32 Bit Local Mask Write Cycle

3.6.2.9 Quad Read Cycle

The timing of the quadruple, 64 bit local read cycle is shown in the figure below:

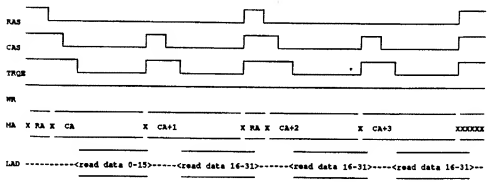


Figure 3.42: Quad, 64 Bit Local Read Cycle

3.6.2.10 Quad Write Cycle

The timing of the quadruple, 64 bit local write cycle is shown in the figure below:

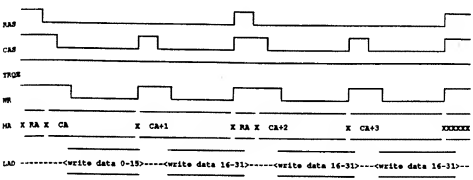


Figure 3.43: Quad, 64 Bit Local Write Cycle

3.6.2.11 Quad Read-modify-write Cycle

The timing of the quadruple, 64 bit local read-modify-write cycle is shown in the figure below:

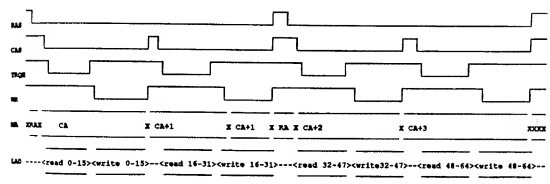


Figure 3.44: Quad, 64 Bit Local Read-modify-write Cycle

3.6.2.12 Quad Mask Write Cycle

The timing of the quadruple, 64 bit local mask write cycle is shown in the figure below:

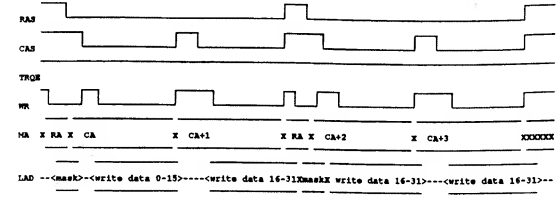


Figure 3.45: Quad, 64 Bit Mask Write Cycle

4 Device Operation

4.1 Operational Modes

4.1.1 Operational Mode Description

Access from the PC host interface to the frame buffer depends upon the the mode of operation of the device. The TMS34096 has four basic modes of operation: text; CGA graphics; 256 colour graphics; and other MDA, EGA and VGA graphics. These can be named modes I, II, III and IV respectively and their correspondance to standard VGA modes 0 to 13 is shown in Table 4.1. Note that since application programs may program register bits to non-standard values, this table represents only a subset of possible modes; non-standard modes are assimilated into operational modes I to IV depending upon a subset of VGA register bits as described in paragraph 4.1.2.

Operational Mode	Corresponding VGA Modes
I	0, 1, 2, 3 & 7
II	4, 5
III	13
IV	6, D, E, F, 10, 11 & 12

Table 4.1: Correspondance Between Operational and VGA Modes

4.1.2 Operational Mode Selection

The selection of operational modes I through IV is determined by the state of three VGA register bits: Memory Mode Register bits 2 (Odd/even) and 3 (Chain4), and Graphics Miscelaneous Register bit 0 (Graphics). The truth table for determining the operations mode depending upon the state of these three bits is shown in Table 4.2.

Operational Mode	Memory Mode b2 (odd/even)	Memory Mode b3 (chain4)	Graphics Misc. b0 (graphics)
I	X*	X	0
II	0	X	1
III	1	1	1
IV	1	0	1

4.1.3 Mode I: Text

The TMS34096 is programmed in operational mode I for any combinaton of VGA register bits where bit 0 of the Graphics Miscelaneous Register is zero. This mode has two sub-modes: text access and font access mode. These are named Sub Mode 1A and 1B respectively. The distinction between the two sub-modes is made by the state of Map Mask Register bit b2 in the case of a write cycle and Map Read Select Register bits b0 and b1 in the case of a read cycle. This is shown in Table 4.3.

Type of Access	Map Mask b2	Read Map Select b1	Read Map Select b0	Sub Mode
Write	0	X*	X	IA
Write	1	X	X	IB
Read	X	X	1	IA
Read	X	0	X	IA
Read	X	1	0	IB

*X indicates a don't care condition for this bit

Table 4.3: Selection of Sub Modes IA and IB

4.1.3.1 Mode IA: Normal Text Access

Mode IA is used by VGA application programs to access character codes and attributes in the frame buffer. Although logically the PC application program views character codes and attributes as being stored in distinct areas of the frame buffer (maps 0 and 1 respectively) the TMS34096 combines these two logical area into one physical area with character codes and attributes interleaved, bytes from PC map 0 being mapped to even bytes of the frame buffer and bytes from map1 to odd bytes. The start of the logical area accessed by the PC application program may be B000:0000, B800:0000 or A000:0000 depending upon bits 2 and 3 of the Graphics Miscelaneous Register as described in paragraph 3.2.6.7. The address of the physical area of memory used to store the text data depend upon the values of bits b0 to b3 of the configuration register **BASE1** and bit b14 of the configuration register **BASE2** as described in paragraph 3.3.2. The actual physical GSP 32 bit address of this area can be represented by: $E_TE_TB_T8$ 0000 where E_T is 0 or F depending upon whether b14 of **BASE2** is 0 or 1 respecively, and B_T is the value of bits b0 through b3 of **BASE1.**

Figure 3.34 shows the mapping from PC space to GSP space and Figure 3.35 the correspondance between PC maps and the real GSP frame buffer.

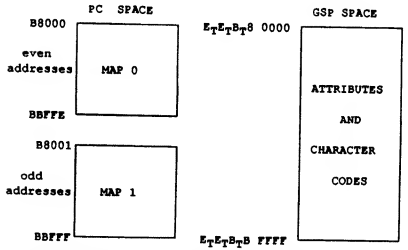


Figure 3.34: Mapping From PC Space to GSP Space in Sub-Mode IA

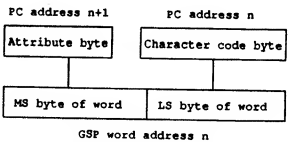


Figure 3.35: Correspondance between PC bytes and GSP words in Sub-mode IA

4.1.3.2 Mode IA: Font Access Mode

The character generator, or font, information is stored in map 2 in a standard VGA system. The TMS34096 therefore detects accesses to this map in either read or write modes and maps the corresponding access to a different part of memory. In addition, in contrast with the normal text access mode where two maps (0 and 1) are interleaved to form one contiguous map, in font access mode one individual map (2)

is mapped directly on a one per one basis into GSP local memory. The base address in GSP local memory is controlled by the same three bits as for attributes and character codes but located exactly 64K bytes below, i.e. the base address of the font map is $E_TE_TB_T0$ 0000 and it extends to $E_TE_TB_T$&7 FFFF.

Figure 3.36 shows the mapping from PC space to GSP space and Figure 3.37 the correspondance between PC maps and the real GSP frame buffer.

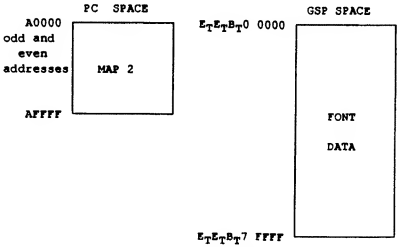


Figure 3.36: Mapping From PC Space to GSP Space in Sub-Mode 1B

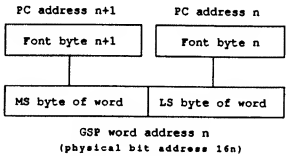


Figure 3.37: Correspondance between PC bytes and GSP words in Sub-mode 1B

4.1.4 Mode II: CGA Graphics

4.1.4.1 Mode II: Description

Operational mode II is selected when bit b0 of the Graphics Miscelaneous Register is set to one and bit b2 of the Memory Mode Register is set to zero. This mode is unlike the other graphics modes in two main ways: firstly the pixel data as seen by the PC application program is in packed pixel format (as opposed to the planar organisation used in all the other graphics modes) and secondly even and odd scan lines are mapped to two different zones in the logical and physical memory space.

4.1.4.2 Mode II: Data Mapping

When the TMS34096 is programmed in mode II, one PC logical byte is mapped into one physical word of local GSP memory as shown in Figure 3.38 below.

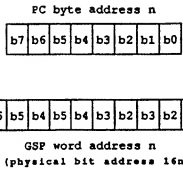


Figure 3.38: Data Mapping in Operational Mode II

The actual mapping of the data during a write access is controlled by the values of bits b0 to b3 of the Map Mask Register. For a bit in memory to be modified its corresponding mask bit needs to be set to one. Map Mask Register b0 masks memory bits 8, 9, 12 & 13, b1 masks bits 0, 1, 4 & 5, b2 masks bits 10, 11, 14 & 15, and b3 masks bits 2, 3, 6 & 7 as shown in Figure 3.39.

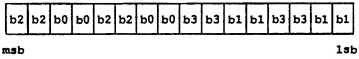


Figure 3.39: Correspondance Between Map Mask and Memory Bits

The mapping of data from GSP to PC is similar for a read cycle with the difference that bits b0 and b0 of the Read Map Select Register control which of the bits in the physical memory word are mapped into the logical PC byte.

4.1.4.3 Mode II: Address Mapping

The start of the logical area accessed by the PC application program may be B000:0000, B800:0000 or A000:0000 depending upon bits 2 and 3 of the Graphics Miscelaneous Register as described in paragraph 3.2.6.7. The address of the physical area of memory used to store Mode II graphics data depends upon the values of bits b4 to b7 of the configuration register **BASE1** and bit b15 of the configuration register **BASE2** as described in paragraph 3.3.2. The actual physical GSP 32 bit address of this area can be represented by: $E_6E_6B_60$ 0000 where E_6 is 0 or F depending upon whether b15 of **BASE2** is 0 or 1 respecively, and B_6 is the value of bits b4 through b7 of **BASE1.**

Figure 3.40 shows the mapping from PC space to GSP space.

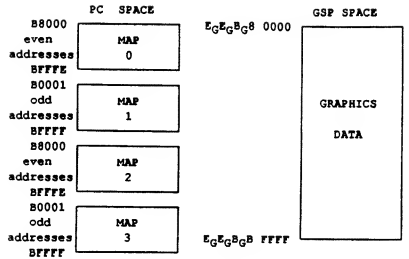


Figure 3.40: Mapping From PC Space to GSP Space in Mode II

4.1.5 Mode III: 256 Colour Graphics

4.1.5.1 Mode III: Description

Operational mode III is selected when bit b0 of the Graphics Miscelaneous Register is set to one and bits b2 and b3 of the Memory Mode Register are set to one. This mode is unlike the other graphics modes in two main ways: firstly the pixel data as seen by the PC application program is in packed pixel format (as opposed to the planar organisation used in nearly all the other graphics modes) and secondly the four logical maps seen by the PC are chained together into one contiguous 64K byte zone.

4.1.5.2 Mode III: Data Mapping

When the TMS34096 is programmed in mode III, two PC logical bytes are mapped into one physical word of local GSP memory as shown in Figure 3.41 below.

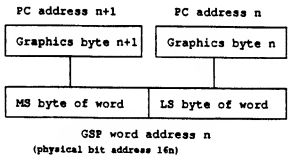


Figure 3.41: Data Mapping in Operational Mode III

4.1.5.3 Mode III: Address Mapping

The start of the logical area accessed by the PC application program may be B000:0000, B800:0000 or A000:0000 depending upon bits 2 and 3 of the Graphics Miscelaneous Register as described in paragraph 3.2.6.7. The address of the physical area of memory used to store Mode III graphics data depends upon the values of bits b4 to b7 of the configuration register **BASE1** and bit b15 of the configuration register **BASE2**

as described in paragraph 3.3.2. The actual physical GSP 32 bit address of this area can be represented by: $E_G E_G B_G 0$ 0000 where E_G is 0 or F depending upon whether b15 of **BASE2** is 0 or 1 respecively, and B_G is the value of bits b4 through b7 of **BASE1.**

Figure 3.42 shows the mapping from PC space to GSP space.

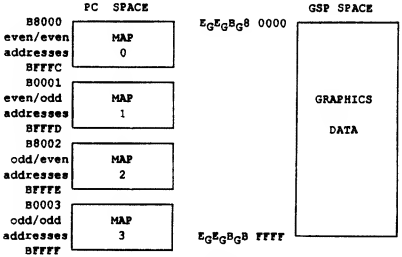


Figure 3.42: Mapping From PC Space to GSP Space in Mode III

4.1.6 Mode IV: Other Graphics Modes

4.1.6.1 Mode IV: Description

Operational mode IV is selected when bit b0 of the Graphics Miscelaneous Register is set to one and bits b2 and b3 of the Memory Mode Register are set to one and zero respectively.

4.1.6.2 Mode IV: Data Mapping

When the TMS34096 is programmed in mode IV, each PC logical byte is mapped into two physical words of local GSP memory as shown in Figure 3.43 below.

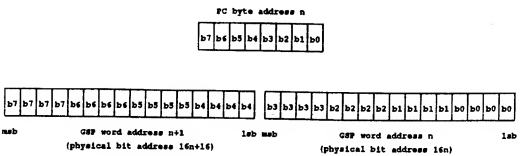


Figure 3.43: Data Mapping in Operational Mode IV

The actual mapping of the data during a write access is controlled by the values of bits b0 to b3 of the Map Mask Register. For a bit in memory to be modified its corresponding mask bit needs to be set to one. Map Mask Register b0 masks memory bits 0, 4, 8 & 12, b1 masks bits 1, 5, 9 & 13, b2 masks bits 2, 6, 10

& 14, and b3 masks bits 3, 7, 11 & 15 as shown in Figure 3.44.

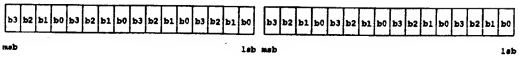


Figure 3.44: Map Masking in Operational Mode IV

4.1.6.3 Mode IV: Address Mapping

The start of the logical area accessed by the PC application program may be B000:0000, B800:0000 or A000:0000 depending upon bits 2 and 3 of the Graphics Miscelaneous Register as described in paragraph 3.2.6.7. The address of the physical area of memory used to store Mode IV graphics data depends upon the values of bits b4 to b7 of the configuration register **BASE1** and bit b15 of the configuration register **BASE2** as described in paragraph 3.3.2. The actual physical GSP 32 bit address of this area can be represented by: $E_GE_GB_GB_G0$ 0000 where E_G is 0 or F depending upon whether b15 of **BASE2** is 0 or 1 respecively, and B_G is the value of bits b4 through b7 of **BASE1.**

Figure 3.45 shows the mapping from PC space to GSP space.

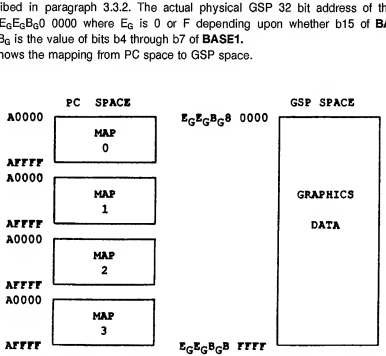


Figure 3.45: Mapping From PC Space to GSP Space in Mode IV

5 Pin assignment

Pin Name	Pin Numbers	Pin description	Type
HCNRDY	001	PC ready signal	O
GND	002,013,018 035,051,068 084,101,117	Ground	F
HCASD2	003	DRAM bank 2 CAS	O
HCASD1	004	DRAM bank 1 CAS	O
HCASD0	005	DRAM bank 0 CAS	O
HCASV1	006	VRAM bank 1 CAS	O
HCASV0	007	VRAM bank 0 CAS	O
DSF	008	VRAM DSF signal	O
VCC	009,014,033 066,099,132	5 volt power	F
HCRTCS	010	CRT I/F chip sel	O
HFLCCS	011	Special chip sel	O
HINTIN	012	34010 interrupt	I
HEXT	015	BIOS extension	O
HHOLD	016	Hold to 34010	O
D0	017	PC dat bus d0	IO
HTRQE	019	VRAM TRQE signal	O
HCAS	020	34010 CAS signal	IO
HRAS	021	34010 & VRAM RAS	IO
HLAL	022	34010 LAL signal	IO
HA0	023	DRAM address a0	O
MA1	024	DRAM address a1	O
MA2	025	DRAM address a2	O
MA3	026	DRAM address a3	O
MA4	027	DRAM address a4	O
MA5	028	DRAM address a5	O
MA6	029	DRAM address a6	O
MA7	030	DRAM address a7	O
MA8	031	DRAM address a8	O
MA9	032	DRAM address a9	O
HIC#16	034	PC IOCH16 signal	O
LAD1	036	34010 LAD bus 1	IO
LAD2	037	34010 LAD bus 2	IO
LAD3	038	34010 LAD bus 3	IO
LAD4	039	34010 LAD bus 4	IO
LAD5	040	34010 LAD bus 5	IO
LAD6	041	34010 LAD bus 6	IO
LAD7	042	34010 LAD bus 7	IO
LAD8	043	34010 LAD bus 8	IO
LAD9	044	34010 LAD bus 9	IO
LAD10	045	34010 LAD bus 10	IO
LAD11	046	34010 LAD bus 11	IO
LAD12	047	34010 LAD bus 12	IO
LAD13	048	34010 LAD bus 13	IO
LAD14	049	34010 LAD bus 14	IO
LAD15	050	34010 LAD bus 15	IO
PCA6	052	PC address A6	I
PCA7	053	PC address A7	I
PCA8	054	PC address A8	I
PCA9	055	PC address A9	I
PCA10	056	PC address A10	I
PCA11	057	PC address A11	I
PCA12	058	PC address A12	I
PCA13	059	PC address A13	I
PCA14	060	PC address A14	I
PCA15	061	PC address A15	I
PCA16	062	PC address A16	I
PCA17	063	PC address A17	I
PCA18	064	PC address A18	I
PCA19	065	PC address A19	I

Pin Name	Pin Numbers	Pin description	Type
HMWRITE	067	VRAM WR signal	O
D1	069	PC data bus D1	IO
D2	070	PC data bus D2	IO
D3	071	PC data bus D3	IO
D4	072	PC data bus D4	IO
D5	073	PC data bus D5	IO
D6	074	PC data bus D6	IO
D7	075	PC data bus D7	IO
D8	076	PC data bus D8	IO
D9	077	PC data bus D9	IO
D10	074	PC data bus D10	IO
D11	079	PC data bus D11	IO
D12	080	PC data bus D12	IO
D13	081	PC data bus D13	IO
D14	082	PC data bus D14	IO
D15	083	PC data bus D15	IO
PCA0	085	PC address A0	I
PCA1	086	PC address A1	I
PCA2	087	PC address A2	I
PCA3	088	PC address A3	I
PCA4	089	PC address A4	I
PCA5	090	PC address A5	I
IO1	091	IO select pin 1	I
HBIOEN	092	BIOS PROM enable	O
IO2	093	IO select pin 2	I
EMBIOS	094	BIOS enable pin	I
CLKIN	095	Master clock in	I
DIR	096	Bus buffar DIR	O
LAD0	097	34010 LAD bus 0	IO
INTOUT	098	PC interrupt out	O
WMC#16	100	PC MEL16 signal	O
HDACK0	102	PC DACK0 signal	I
HIOR	103	PC IOR signal	I
HIOW	104	PC IOW signal	I
HSMEMR	105	PC SMEMR signal	I
HSMEMW	106	PC SMEMW signal	I
RST	107	Device reset	I
HSEL16	108	16 bit enable	I
SCLK	109	VRAM SCLK signal	I
HREN	110	PC MEMR signal	I
HVSYNC	111	Vartical sync	I
HBLANK	112	Composita blank	I
HRDY	113	34010 ready	I
HHOLDA	114	34010 hold ack	I
TEST	115	Test control	I
LCLK2	116	34010 LCLK2	I
HENL	118	Low buffar enbl.	O
HEMM	119	Medium buff.enbl	O
HEMH	120	High buffar enbl	O
HRESET	121	34010 reset	O
HF1	122	34010 F1 signal	O
HF0	123	34010 F0 signal	O
HHCS	124	34010 HCS signal	O
HHWR	125	34010 HWR signal	O
HHREAD	126	34010 HREAD sig	O
HHLDS	127	34010 HLDS sig	O
HHUDS	128	34010 HUDS sig	O
HPALWR	129	Palette write	O
HPALRD	130	Palette read	O
HCASD3	131	DRAM bank 3 CAS	O

Appendix I.

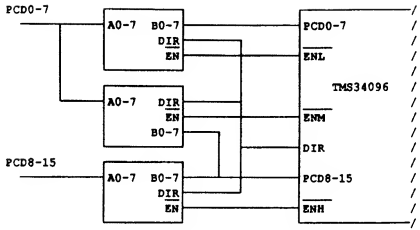


Figure I.1: PC Interface Bus Buffering

Claims

1. Display adapter connected between a host processor (2) and a display unit (6) of a computerized tool wherein a graphics processor (1) is connected between the host processor and a first part of a memory (4) associated with the display unit (6) and a logic based hardware sub-system (7) is connected between said host processor (2) and a second part of said memory (4) and comprising means for deriving displays from the first and/or the second parts of the memory.

2. Display adapter according to claim 1, wherein the deriving means comprises means (1a) for transferring the data stored in the second part (8) of the video memory in the first part (3) of it.

3. Display adapter according to claim 2, wherein the transferring means comprises means for merging in the first part of the memory, the data stored in this first part and in the second part of the memory.

4. Display adapter according to claim 3, wherein the merging means (1a) comprises means for copying data from the second part (8) of the video memory in the first part of it.

5. Display adapter according to claim 4, wherein the merging means (1a) comprises a subroutine of the graphics processor (1).

6. Display adapter according to one of the preceding claims, wherein the logic based hardware sub-system (7) comprises :
   - a PC bus interface (20) receiving as inputs PC control, PC address and PC data signals from the host processor (2),
   - a sequence controller (2), connected between a corresponding output of the PC bus interface (20) and a display controller (25),
   - a GSP/LAD bus interface (26) connected between the display controller (25) and a corresponding input of a memory controller (28) and to a LAD bus,
   - an address mapper (22) and a data mapper (23) connected between corresponding outputs of the PC bus interface (20) and corresponding inputs of an access arbitration controller (27) also connected at an output of the GSP/LAD bus interface (26) and two outputs of which are connected to corresponding inputs of the memory controller (28), and
   - internal registers (24) connected to the outputs of the address and data mappers (22, 23).

7. Display adapter as claimed in any preceeding claim and wherein said memory is video memory.

8. Display adapter as claimed in any preceeding claim and including an input for VGA type display data.

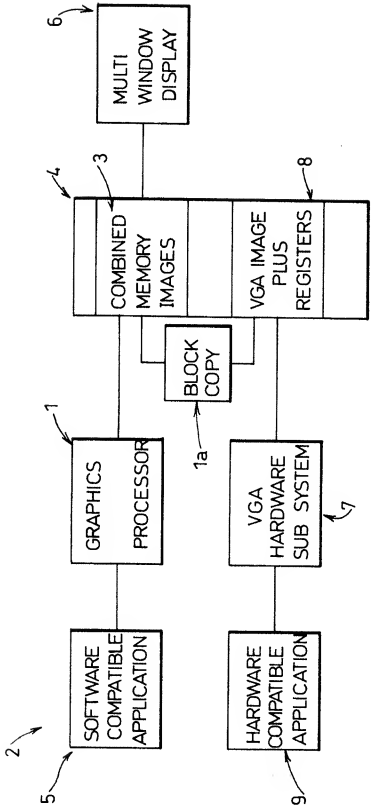
9. Display system including a display adapter as claimed in any preceeding claim.

10. Display system as claimed in claim 9 and wherein VGA type data is manipulated by the graphics signal processor.

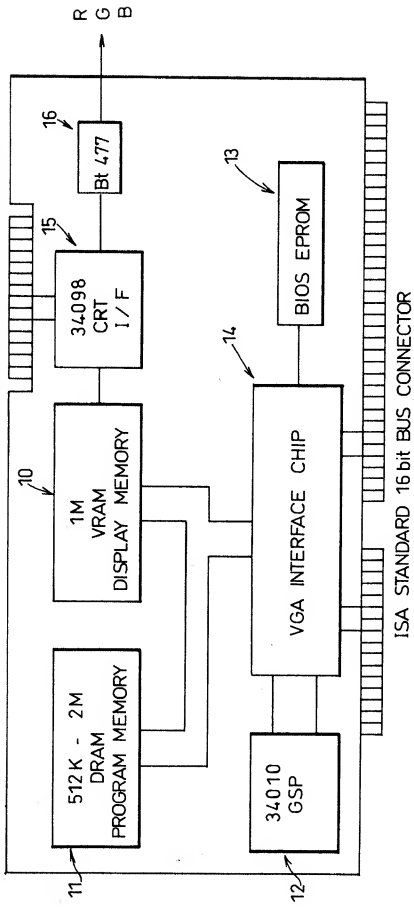
11. Method of providing a display including the steps of storing graphics processor type display data to a first part of memory associated with a display unit, storing VGA type display data to a second part of memory and deriving a display from the first and/or the second part of the memory.

12. Method as claimed in claim 11 and wherein data from the second part of memory is manipulated by a graphics signal processor.

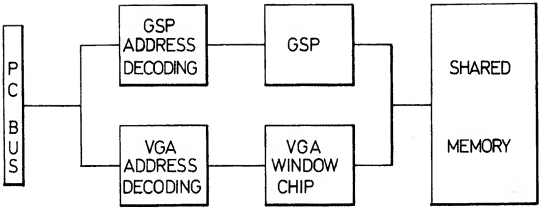
13. Computerized tool comprising a display adapter as claimed in claim 9 or 10.


6
MULTI WINDOW DISPLAY
4
3
COMBINED MEMORY IMAGES
8
VGA IMAGE PLUS REGISTERS
BLOCK COPY
1a
1
GRAPHICS PROCESSOR
VGA HARDWARE SUB SYSTEM
7
2
5
SOFTWARE COMPATIBLE APPLICATION
HARDWARE COMPATIBLE APPLICATION
9

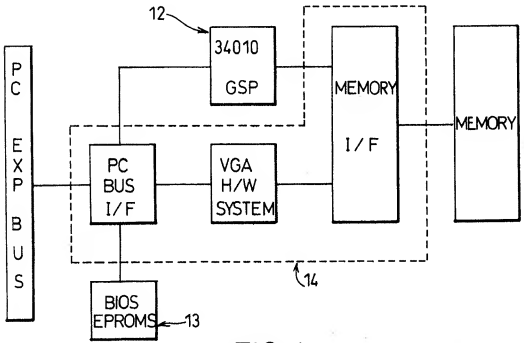
FIG·1


R
G
B
VGA FEATURE CONNECTOR
16
Bt 477
15
34098
CRT
I / F
13
BIOS EPROM
14
10
1M
VRAM
DISPLAY MEMORY
VGA INTERFACE CHIP
512 K - 2M
DRAM
PROGRAM MEMORY
34010
GSP
11
12
ISA STANDARD 16 bit BUS CONNECTOR

FIG·2


PC BUS
GSP ADDRESS DECODING
GSP
SHARED MEMORY
VGA ADDRESS DECODING
VGA WINDOW CHIP

FIG·3


12
34010 GSP
MEMORY I/F
MEMORY
PC EXP BUS
PC BUS I/F
VGA H/W SYSTEM
14
BIOS EPROMS
13

FIG·4

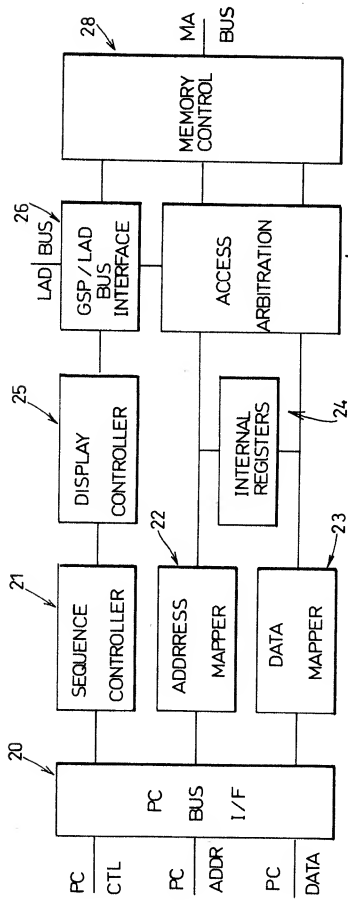

PC CTL
PC ADDR
PC DATA
PC BUS I/F
20
SEQUENCE CONTROLLER
21
ADDRRESS MAPPER
22
DATA MAPPER
23
DISPLAY CONTROLLER
25
INTERNAL REGISTERS
24
ACCESS ARBITRATION
27
LAD BUS
GSP / LAD BUS INTERFACE
26
MEMORY CONTROL
28
MA BUS

FIG. 5

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 91 40 2072

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
A	EP-A-0 419 814 (I.B.M. CO.) 3 April 1991 * Abstract * * column 1, line 1 - line 9; figures 4,6 * * column 7, line 7 - line 45 *	1,8,9, 11,13	G06F3/14
A	EP-A-0 396 377 (EVANS & SUTHERLAND COMPUTER CO.) 7 November 1990 * page 3, line 43 - page 5, line 10; figure 1 *	1,9,11, 12	
D,A	US-A-4 752 893 (TEXAS INSTRUMENTS INC.) 21 June 1988 * Abstract * * figure 1 *	1,6,7,9, 11	

TECHNICAL FIELDS SEARCHED (Int. Cl.5)

G06F
G09G

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	25 MARCH 1992	CORSI F.

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P0401)